Report on Project 2a: Alloy cluster expansions

Alfred Juhlin Onbeck and Zhi Li

March 25, 2024

Task 1

Discuss: Why is it good practice to standardize the data? Is it necessary in this case?

Standardising the data by giving the cluster vectors and mixing energies unit mean and variance, and saving the previous mean, standard deviation to later transform back to original energy scale.

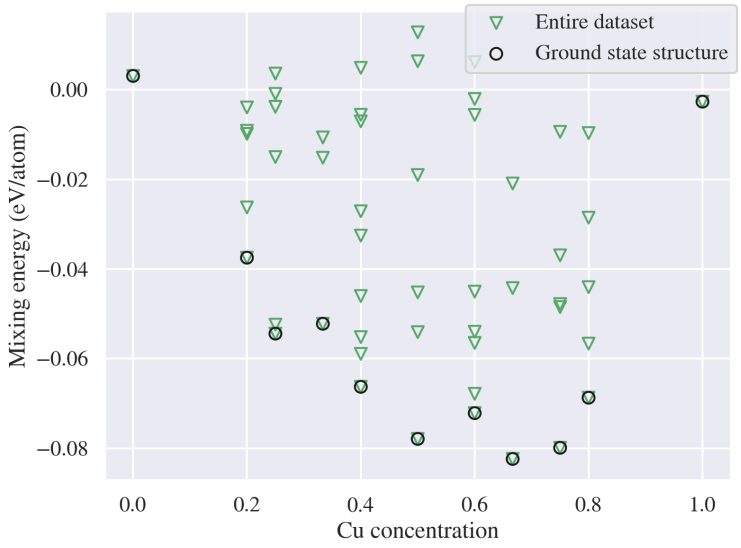

Figure 1: `Au_Cu_structures` database. Visualizing mixing energy for different Cu concentrations, structures. The ground state for each Cu concentration is represented by a black circle, and the entire dataset is shown as green triangles.

Standardization ensures that each feature contributes equally to the analysis, regardless of their original scale.[1] Standardization helps in maintaining a consistent scale across all dimensions, which facilitates smoother and faster optimization.

Standardization is important in this case. By standardizing, we make sure that the regularization is evenly distributed. If features are on different scales, the regularization term does not apply uniformly across them.This can lead to an imbalance in the regularization process.

Task 2

Discuss: What are the qualitative differences between OLS and Ridge?

Ridge regression Eq. 1 and ordinary least squares are linear regression methods that aim to minimize the squared differences between predictions and observed data.

$$J_{opt,Ridge} = (X^T X + \alpha I)^{-1} X^T E. \tag{1}$$

In order to properly evaluate the models performance 10-fold cross validation has been used for training all models and seperate data for final evaluation. Here we compare the models root-mean-square of the testing data for the two linear models, both implemented through `sklearn.linear_model`. For the CV OLS, RMSE: 0.1664, and of CV ridge regression ($\alpha = 0.0756$): 0.1497. Where α was an additional parameter that was optimized. Ridge regression is similar to OLS with the additional penelization α for large coefficients, this can be beneficial for improving generalization and appears to be outperforming the OLS model.

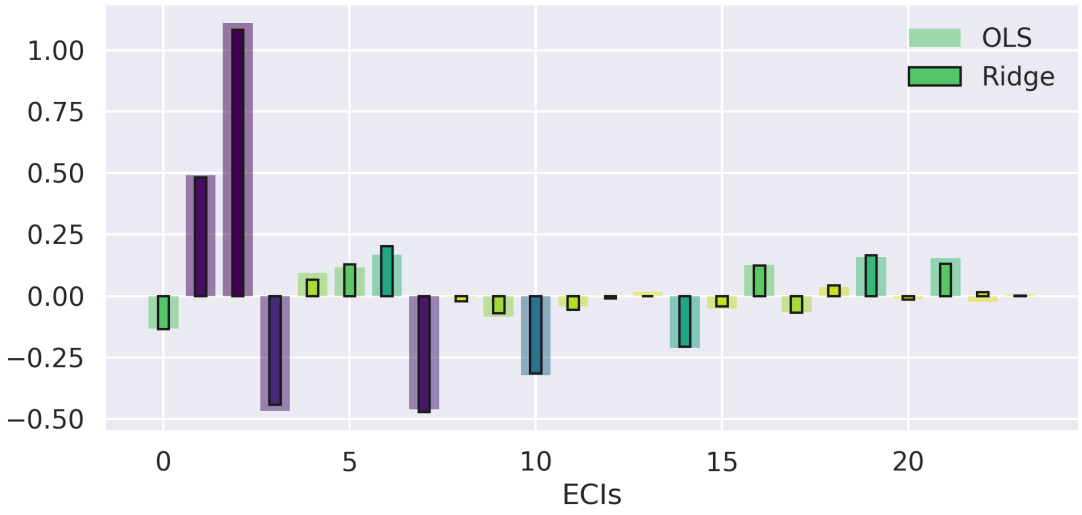

Figure 2: Visualization of the model, ECIs. Here the narrow box with a black outline is the Ridge model and the transparent wider boxes is the OLS model. The boxes color correspond to how close they are to zero with purple color for farthest from zero and yellow for close to zero. For most parameters the models are indistinguishable. Small differences can be seen for parameters 4,5,6.

OLS: Its goal is to identify the most accurate line that minimizes the total squared differences between the actual data points and the values predicted by the linear model.

Ridge Regression: Utilized within linear regression, it tackles the issue of overfitting. It achieves this by introducing a regularization component to the loss function. This component causes the model's coefficients to shrink towards zero,[2] thereby diminishing the model's variance and potentially enhancing its ability to predict.

Task 3

Discuss: What is the interpretation of having an individual parameter λ_α for each orbit? How does this compare to Ridge regression? How does the CV-RMSE and the ECIs compare to OLS and Ridge? Can you explain it?

Similarily to Ridge regression, this model will have hyperparameters to penalize certain cluster orbits. The optimal ECIs J can be determined by Eq. 2,

$$J_{opt,Cov} = \left(X^T X + \Lambda\right)^{-1} X^T E, \tag{2}$$

with Λ being a diagonal matrix with elements $\lambda_\alpha = \gamma_1 r + \gamma_2 n$. This penalizes ECIs J with larger radius and more sites which coincides with our prior knowledge of cluster orbits. To find the optimal hyperparameters γ_1, γ_2 we used the cross validated RMSE as a loss function and let `scipy.optimize.minimize` do our optimization. This resulted in a minimum of CV RMSE $= 0.1211$, with $\gamma_{1,2} = [-0.0116, 0.0204]$.

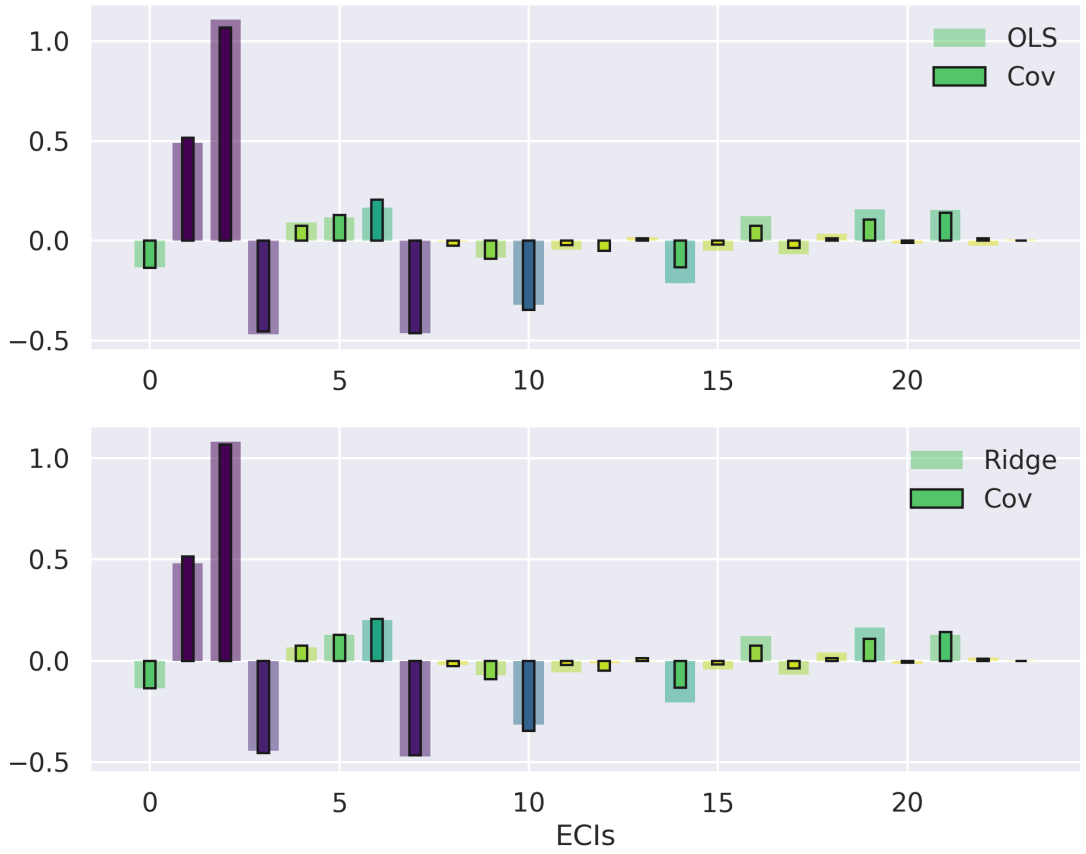

Figure 3: Comparing ECIs for three different models. Here the narrow box with a black outline is the Cov model and the transparent wider boxes with no edge color is the OLS model for upper figure, Ridge for the lower figure. The boxes color correspond to how close they are to zero with purple color for farthest from zero and yellow for close to zero.

Task 4

Discuss: How many parameters seems "necessary" (non-zero) according to your MCMC sampling? What would happen if you had set your priors to something "unphysical", for instance to favor 3rd and 4th order clusters very highly?

For a Bayesian analysis of the effective cluster interactions we need to define our priors and a likelihood function. The prior used for ECIs J

$$P(\boldsymbol{J}) = \frac{1}{(2\pi\alpha^2)^{\boldsymbol{N}_p/2}} \exp\left(-||\boldsymbol{J}||^2/2\alpha^2\right), \tag{3}$$

and for σ, α inverse-gamma priors were used and implemented with `scipy.stats.invgamma`. Since we know roughly that σ should be in $[0.02, 2]$ and α in $[0.05, 0.5]$ we let $\alpha_\sigma = 2, \beta_\sigma = 1$ and $\alpha_\alpha = 5, \beta_\alpha = 1$, so that they cover their corresponding ranges more or less. The likelihood function that we used is a multivariate Gaussian Eq. 4, and the posterior is (Eq. 5) defined as the product of the likelihood and the priors, if you disregard the marginal likelihood.

$$L = P(D|\boldsymbol{J}, \sigma) = \frac{1}{(2\pi\sigma^2)^{\mathbf{V}_{d'2}}} \exp\left(-||\boldsymbol{X}\boldsymbol{J} - \boldsymbol{E}||^2/2\sigma^2\right). \tag{4}$$

$$P(\boldsymbol{J}, \sigma^2, \alpha \mid \boldsymbol{D}) = P(\boldsymbol{D} \mid \boldsymbol{J}, \sigma^2) P(\boldsymbol{J}, \sigma, \alpha). \tag{5}$$

To sample our posterior we used `emcee` with 400 walkers. The walkers initial values for $\boldsymbol{J}$ was based on the value from $J_{opt,Cov}$ with some Gaussian noise added, for α, σ two uniform distribution were used for their respective ranges that was mentioned before. The initial values for α, σ are a bit too improbable which means we will had to disregard the first 150 iterations due to burn in, this was clear after visualizing the first 500 iterations for a couple of chains. To obtain uncorrelated samples we need to determine out effective sample size (ESS), this has implemented with `arviz.ess` has gave us roughly 58 uncorrelated models per chain, out of 20 000 total iterations per chain. Based on fig. 7 it appears that 20 non-zero parameters would suffice. If our priors were based something unphysical the posterior would most likely be biased with poor genaralization and ultimately bad inferences.

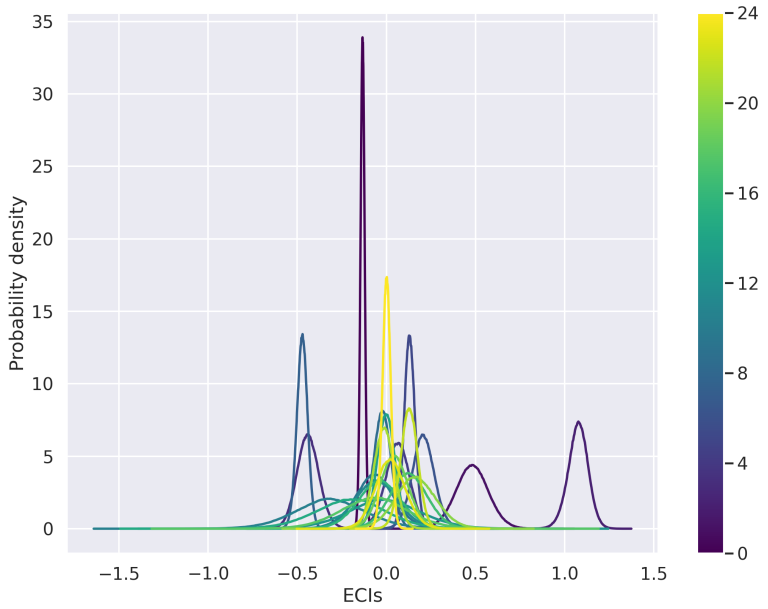

Figure 4: ECIs of MCMC sampled models.

Posterior distribution of α, σ can be seen in fig. 5.

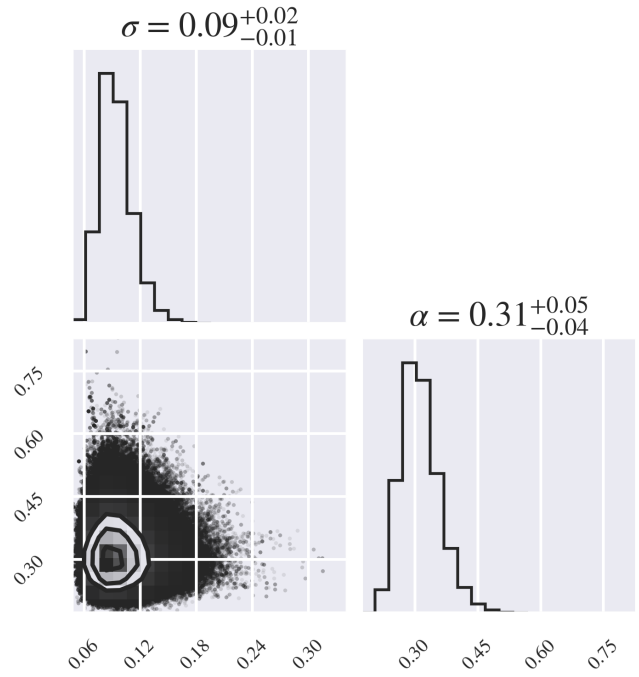

Figure 5: Corner plot showing posterior distrubtion of α, σ and their joint posterior.

Task 5

Discuss: How many features you think is suitable to include in a final model based on your analysis. Which ECIs are selected here? Is there a difference to OLS/Ridge/Covariance approach?

Automatic Relevance Detection (ARD) Regression is more of a automatic approach towards fitting a model than the previous ways. Our implementation will be using `sklearn.linear_model.ARDRegression` while only optimizing the `lambda_threshold` parameter. In fig. 6 a sweep of λ-threshold is shown where the mean-absolute-error and the IC scores are displayed. If the absolute value of a parameter is lower than 10^{-2} then we do not consider it a parameter when calculating the information criteria (IC) scores.

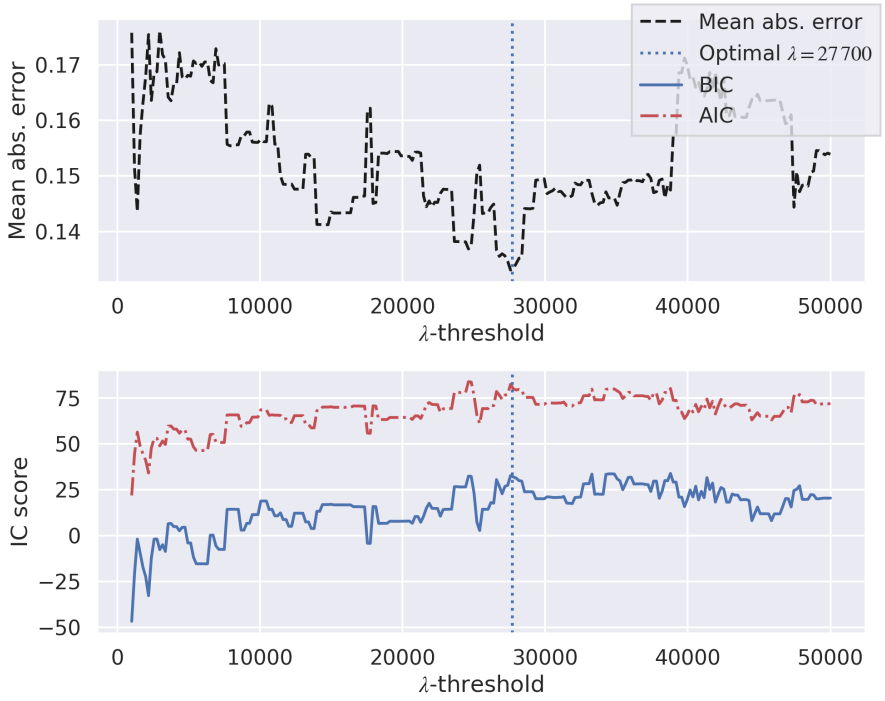

Figure 6: Optimization of the λ-threshold parameter. Around 27 700 is ideal for AIC, BIC scores and low error.

When comparing the ARD model to the other ones it is clear that it is a less complex model with 6-8 fewer parameters, while not having that much higher RMSE. In table 1 the Cov model has the highest AIC score, N_p and lowest RMSE while ARD has the fewest parameters and highest BIC score. MAP is the maximum of the MCMC sampled posterior distribution. All models were trained on 10-fold cross validated data, seperate from the testing data.

	RMSE	AIC	BIC	N_p
OLS	0.17	105.0	69.0	21
Ridge	0.15	111.7	74.0	22
Cov	0.12	128.2	90.5	22
MAP	0.18	102.7	68.4	20
ARD	0.18	118.2	94.2	14

Table 1: Comparison of the models based on testing data from `Au_Cu_structures.db`

In fig. 7 it is clear that the ARD model has fewer nonzero parameters in comparison to the MAP model.

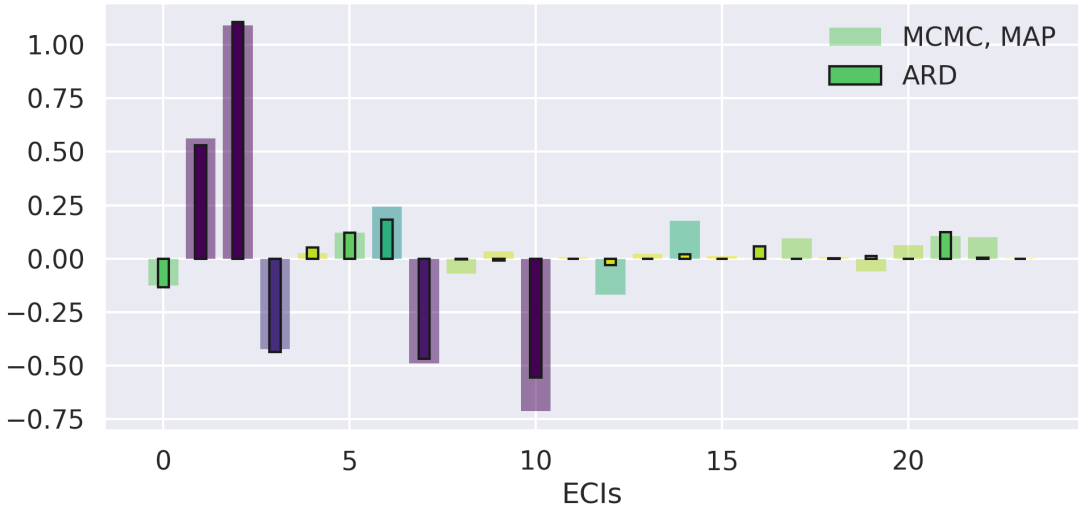

Figure 7: Comparing ECIs of ARD model and the maximum of the sampled posterior distribution. The sampled model is represented by transparent larger boxes and ARD by smaller, outlined boxes.

Based on table 1, somewhere between 14-20 seems adequate where 14 might be on the verge of underfitting. For OLS, Ridge and Cov, all share the same ECIs more or less with MAP and ARD models being the outliers. This can be seen in the figures 2, 3, 7, where e.g MCMC, MAP is the only model with the 2nd to last ECI being non-zero. All models also seem almost equivalent for the first seven features while the noticable deviations being are around feature ten and after.

Task 6

Here we will be considering structures with 0.667 Cu concentration and predicting their energy to find the ground state structure. The uncorrelated models from the MCMC sampling will be used as a reference for comparing the models. In fig. 8 it is shown that all models managed to predict the same structure to be the ground state. There is also no overlap among the sampled model distributions which means there is 100% certainty which is the ground state structure (structure with index 5 in the `ground_states_candidates.db`).

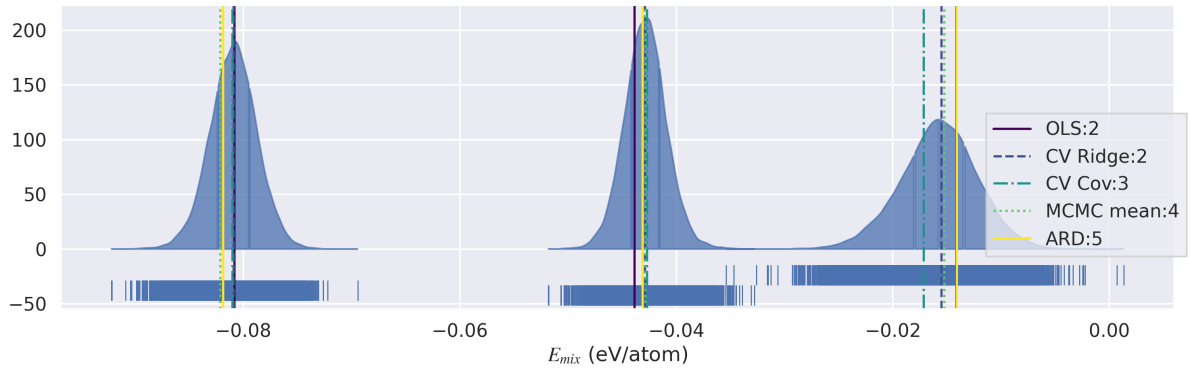

Figure 8: The three different structures with C_{Cu} of 0.667 with one being the clear ground state structure. The energy distributions correspond to the MCMC sampled models that show an interval where the mixing energy should be.

Therefore we consider the ARD regression model to be the best approach since it does an adequate job in inferring the mixing energy of the structures while being considerably less complex than the other models, while also being mostly autonomous when optimizing.

References

[1] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, January 2006. [Online]. Available: https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/

[2] sumanthkuchipudi838. (2023) Ordinary least squares and ridge regression variance in scikit learn. 1.12.2023. [Online]. Available: https://www.geeksforgeeks.org/ordinary-least-squares-and-ridge-regression-variance-in-scikit-learn/